

ליאור פרץ

ראובן קליין

דו"ח מיני פרויקט 2

השיפור הנבחר: Adaptive Super Sampling

רקע:

במיני פרויקט 1 ביצענו את השיפור Glossy surfaces and diffused glass. לצורך כך יצרנו משטח מטרה אליו שלחנו קרניים בפיזור סביב קרן ההשתקפות/שקיפות וחישבנו את ממוצע הצבעים המתקבל מן כל הקרניים.

כדי לקבל תמונה איכותית יש לשלוח מספר רב של קרניים, דבר המאט מאוד את יצירת התמונה. כדי לשפר את זמן הריצה ניתן להשתמש ב-Adaptive super sampling כדי לדגום את משטח המטרה במקום לשלוח קרניים אל כולו מראש.

ביצוע:

הדרך לביצוע היא לשלוח קרניים לארבעת הקודקודים של המשטח ולבדוק האם הצבע שלהם קרוב מספיק. אם כן, מניחים שכל המשטח בצבע הזה ומחזירים את הממוצע שלהם. אם לא, מחלקים את המשטח ל-4 ריבועים חדשים ושולחים קרניים לארבעת הקודקודים. הצבע שיוחזר יהיה הצבע הממוצע המתקבל מארבעת הריבועים.

כך ממשיכים בצורה רקורסיבית עבור כל אחד מן הריבועים, עד שמגיעים לחלוקה המקסימלית שהוגדרה מראש.

כדי למנוע מצב של שליחת קרניים כפולות, שומרים את הצבעים המתקבלים מן כל קרן, ולפני ששולחים קרן בודקים האם הצבע כבר קיים במאגר.

קוד:

ראשית, ביצענו refactoring לשלב הקודם. יצרנו בחבילה Render מחלקה בשם TargetSurface שמייצגת את משטח המטרה באמצעות הנקודה השמאלית העליונה שלו, וקטור ימינה, וקטור שמאלה, אורך הצלע שלו וגודל הריבועים אליהם מחלקים את המשטח.

כמו כן שינינו את הפונקציה calcColorFromBeamOfRays כך שכעת היא יוצרת אובייקט של משטח מטרה, ומזמנת אחת מן שתי הפונקציות- colorFromRegularBeam במידה והפיצ'ר של adaptive super sampling מושבת או את הפונקציה adaptiveSuperSampling.

הפונקציה `adaptiveSuperSampling` היא פונקציה עוטפת לפונקציה רקורסיבית נוספת בשם זה. בפונקציה זו יוצרים מערך דו מימדי ריק של `Color` ומזמנים את הפונקציה הרקורסיבית עבור ארבעת הפינות של משטח המטרה.

בפונקציה הרקורסיבית `adaptiveSuperSampling` ראשית בודקים האם הצבע של הנקודות שהתקבלו כבר נמצא במטריצה, ואם לא, מחשבים אותו באמצעות הפונקציה החדשה `calcColor` שמזמנת את `calcGlobalEffect` עבור הקרן לנקודה.

לאחר שיש ערכים לארבעת הצבעים, בודקים האם הם מספיק דומים (באמצעות הפונקציה `Color.equals()`) שעשינו לה ההעמסה שתחזיר `true` עבור פער בין צבעים הקטן ממשתנה במחלה `Color` בשם `MIN_COLORS_DIFFERENCE`.

במידה והצבעים מספיק דומים מוחזר הצבע הממוצע שלהם. אם לא, בודקים האם הגענו כבר לשלב הסופי של הרקורסיה (כל עוד המשתנה `MAX_SUPER_SAMPLING_LEVEL` גדול מ-1 ממשיכים). אם הגענו לשלב הסופי מחזירים את הממוצע של ארבעת הצבעים. אם לא, מחלקים את הריבוע ל-4 ריבועים קטנים ומזמנים את `adaptiveSuperSampling` עבור כל אחד מהם.

תוצאה:

עבור התמונה המצורפת, זמן הריצה עם 3 threads ו-1089 קרניים ללא `super sampling` לקח כשעתיים וחצי.

עבור ריצה עם 3 threads ועומק `MAX_SUPER_SAMPLING_LEVEL=6` (מקביל ל-1089 קרניים במקרה הגרוע) זמן הריצה היה 9 דקות.

